

An Algorithm for Road Coloring

A.N. Trahtman*

Bar-Ilan University, Dep. of Math., 52900, Ramat Gan, Israel

Abstract. A coloring of edges of a finite directed graph turns the graph into finite-state automaton. The synchronizing word of a deterministic automaton is a word in the alphabet of colors (considered as letters) of its edges that maps the automaton to a single state. A coloring of edges of a directed graph of uniform outdegree (constant outdegree of any vertex) is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

The road coloring problem is the problem of synchronizing coloring of a directed finite strongly connected graph of uniform outdegree if the greatest common divisor of the lengths of all its cycles is one. The problem posed in 1970 had evoked a noticeable interest among the specialists in the theory of graphs, automata, codes, symbolic dynamics as well as among the wide mathematical community.

A polynomial time algorithm of $O(n^3)$ complexity in the most worst case and quadratic in majority of studied cases for the road coloring of the considered graph is presented below. The work is based on recent positive solution of the road coloring problem. The algorithm was implemented in the package TESTAS (<http://www.cs.biu.ac.il/~trakht/syn.html>)

Keywords: algorithm, road coloring, graph, deterministic finite automaton, synchronization

Introduction

The road coloring problem was stated almost 40 years ago [Adler, Weiss 70], [Adler et al. 77] for a strongly connected directed finite deterministic graph of uniform outdegree where the greatest common divisor (gcd) of the lengths of all its cycles is one. The edges of the graph being unlabelled, the task is to find a labelling of the edges that turns the graph into a deterministic finite automaton possessing a synchronizing word. The outdegree of the vertex can be considered also as the size of an alphabet where the letters denote colors.

The condition on gcd is necessary [Adler et al. 77], [Culik et al. 01]. It can be replaced by the equivalent property that there does not exist a partition of the set of vertices on subsets $V_1, V_2, \dots, V_k = V_1$ ($k > 2$) such that every edge which begins in V_i has its end in V_{i+1} [Culik et al. 01], [O'Brien 81].

Together with the Černy conjecture [Černy 64], [Mateescu, Salomaa 99], the road coloring problem used to belong to the most fascinating problems in the

* Email: trakht@macs.biu.ac.il

theory of finite automata. The popular Internet Encyclopedia "Wikipedia" mentions it on the list of most interesting unsolved problems in mathematics.

For some results in this area see [Budzban et al. 0], [Carbone 01], [Friedman 1990], [Gocka et al. 98], [Jonoska et al 1995], [Kari 01], [O'Brien 81], [Perrin et al. 92], [Hegde et al. 05], A detailed history of investigations can be found in [Carbone 01]. The final positive solution of the problem stated in [Trahtman 09].

An algorithm for road coloring oriented on DNA computing [Jonoska et al. 98] is based on massive parallel computing of sequences of length $O(n^3)$. The implementation of the algorithm as well as the implementation of effective DNA computing is still a problem of future.

An another new algorithm for road coloring based on a preprint of [Trahtman 09] and skilful study of the graph was supposed in ArXiv [Béal, Perrin 08]. The algorithm was declared as quadratic, however some gaps in the proofs must be removed before publication and implementation.

Our algorithm for the road coloring announced in [Trahtman 07] also is based on the recent positive solution of the problem. The concept of a stable pair of states from [Culik et al. 01], [Kari 01] is used throughout the work. The theorems and lemmas from [Trahtman 09] and [Trahtman 08b] are presented below without proof. The proofs are given only for new or modified results. The direct use of the last papers ensures only cubic time complexity. However, more detailed study gives us opportunity to reduce quite often the complexity.

The time and space complexity of the algorithm for graph with n vertices and d outgoing edges of any vertex is $O(n^3d)$ in the most worst case and quadratic in majority of the studied cases. The algorithm is implemented in the package TESTAS [Trahtman 06]. The realization of the algorithm is demonstrated on the base of a linear visualization program [Bauer et al. 07] and can analyze any kind of input graph. Today it is the best implemented algorithm.

The role of the road coloring and the algorithm is substantial also in education because the problem can be stated in simple terms and initial explorations can be done immediately. "The Road Coloring Conjecture makes a nice supplement to any discrete mathematics course" [Rauff 09].

Preliminaries

As usual, we regard a directed graph with letters assigned to its edges as a finite automaton, whose input alphabet Σ consists of these letters. The graph is called *transition graph* of the automaton. The letters from Σ can be considered as colors and the assigning of colors to edges will be called coloring.

A finite directed strongly connected graph with constant outdegree of all its vertices where the gcd of lengths of all its cycles is one will be called *AGW graph* as aroused by Adler, Goodwyn and Weiss.

We denote by $|P|$ the size of the subset P of states of an automaton (of vertices of a graph).

If there exists a path in an automaton from the state \mathbf{p} to the state \mathbf{q} and the edges of the path are consecutively labelled by $\sigma_1, \dots, \sigma_k$, then for $s = \sigma_1 \dots \sigma_k \in \Sigma^+$ we shall write $\mathbf{q} = \mathbf{p}s$.

Let P_s be the set of states $\mathbf{p}s$ for $\mathbf{p} \in P$, $s \in \Sigma^+$. For the transition graph Γ of an automaton let Γs denote the map of the set of states of the automaton.

A word $s \in \Sigma^+$ is called a *synchronizing* word of the automaton with transition graph Γ if $|\Gamma s| = 1$.

A coloring of a directed finite graph is *synchronizing* if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

The bold letters will denote the vertices of a graph and the states of an automaton.

A pair of distinct states \mathbf{p}, \mathbf{q} of an automaton (of vertices of the transition graph) will be called *synchronizing* if $\mathbf{p}s = \mathbf{q}s$ for some $s \in \Sigma^+$. In the opposite case, if for any s $\mathbf{p}s \neq \mathbf{q}s$, we call the pair a *deadlock*.

A synchronizing pair of states \mathbf{p}, \mathbf{q} of an automaton is called *stable* if for any word u the pair $\mathbf{p}u, \mathbf{q}u$ is also synchronizing [Culik et al. 01], [Kari 01].

We call the set of all outgoing edges of a vertex a *bunch* if all these edges are incoming edges of only one vertex.

The subset of states (of vertices of the transition graph Γ) of maximal size such that every pair of states from the set is a deadlock will be called an *F-clique*.

1 Some properties of *F*-cliques and of stable pairs

The road coloring problem was formulated for *AGW* graphs [Adler et al. 77] and only such graphs are considered below.

Let us recall that a binary relation ρ on the set of the states of an automaton is called congruence if ρ is equivalence and for any word u from $\mathbf{p} \rho \mathbf{q}$ follows $\mathbf{p}u \rho \mathbf{q}u$. Let us formulate an important result from [Culik et al. 01], [Kari 01] in the following form:

Theorem 1 [Kari 01] *Let us consider a coloring of an AGW graph Γ . The stability of states is a binary relation on the set of states of the obtained automaton. Let us denote this relation by ρ . Then ρ is a congruence relation, Γ/ρ is an AGW graph and a synchronizing coloring of Γ/ρ implies synchronizing recoloring of Γ .*

Lemma 1 [Trahtman 09] *Let F be an *F*-clique of some coloring of an AGW graph Γ . For any word s the set Fs is also an *F*-clique and any state \mathbf{p} belongs to some *F*-clique.*

Lemma 2 *Let A and B (with $|A| > 1$) be distinct *F*-cliques of some coloring of an AGW graph Γ such that $|A| - |A \cap B| = 1$. Then for all $\mathbf{p} \in A \setminus A \cap B$ and $\mathbf{q} \in B \setminus A \cap B$ the pair (\mathbf{p}, \mathbf{q}) is stable.*

Proof. By the definition of an *F*-clique, $|A| = |B|$ and $|B| - |A \cap B| = 1$, too. If the pair of states $\mathbf{p} \in A \setminus B$ and $\mathbf{q} \in B \setminus A$ is not stable, then, for some

word s the pair $(\mathbf{p}s, \mathbf{q}s)$ is a deadlock. Any pair of states from the F -clique A and from the F -clique B , as well as from the F -cliques As and Bs , is a deadlock. So any pair of states from the set $(A \cup B)s$ is a deadlock. One has $|(A \cup B)s| = |As| + 1 = |A| + 1 > |A|$. So the size of the set $(A \cup B)s$ of deadlocks is greater than the maximal size of F -clique. Contradiction.

Lemma 3 *If some vertex of AGW graph Γ has two incoming bunches then the origins of the bunches form a stable pair by any coloring.*

Proof. If a vertex \mathbf{p} has two incoming bunches from \mathbf{q} and \mathbf{r} , then the couple \mathbf{q}, \mathbf{r} is stable for any coloring because $\mathbf{q}\sigma = \mathbf{r}\sigma = \mathbf{p}$ for any $\sigma \in \Sigma$.

2 The spanning subgraph of AGW graph

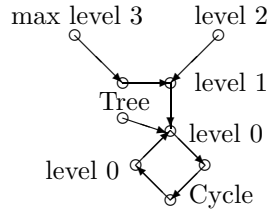
Definition 4 *Let us call a subgraph S of an AGW graph Γ , a spanning subgraph of Γ , if S contains all vertices of Γ and if each vertex has exactly one outgoing edge. (In usual graph-theoretic terms it is 1-outregular spanning subgraph).*

A maximal subtree of a spanning subgraph S with its root on a cycle from S and having no common edges with the cycles of S is called a tree of S .

The length of path from a vertex \mathbf{p} through the edges of the tree of the spanning set S to the root of the tree is called a level of \mathbf{p} in S .

A tree with vertex of maximal level let us call a maximal tree.

Remark 5 *Any spanning subgraph S consists of disjoint cycles and trees with roots on the cycles. Any tree and cycle of S is defined identically. The level of the vertices belonging to some cycle is zero. The vertices of the trees except the roots have positive level. The vertices of maximal positive level have no incoming edge in S . The edges labelled by a given color defined by any coloring form a spanning subgraph. Conversely, for each spanning subgraph, there exists a coloring and a color such that the set of edges labelled with this color corresponds to this spanning subgraph.*



Lemma 6 [Trahtman 09] [Trahtman 08b] *Let N be a set of vertices of maximal level in some tree of the spanning subgraph S of an AGW graph Γ . Then, via a coloring of Γ such that all edges of S have the same color α , for any F -clique F holds $|F \cap N| \leq 1$.*

Lemma 7 [Trahtman 09] *Let Γ be an AGW graph with a spanning subgraph R which is union of cycles (without trees). Then the non-trivial graph Γ has another spanning subgraph with exactly one maximal tree.*

Lemma 8 *Let us assume that any vertex of an AGW graph Γ has not two incoming bunches. Let R be a spanning subgraph of Γ . Let T be a maximal tree with a vertex \mathbf{p} of maximal positive level and let \mathbf{r} be its root which belongs to a cycle H of R . Let us consider the following flips*

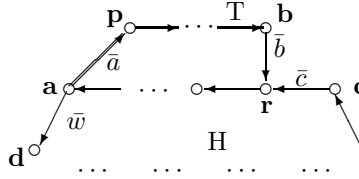
- 1) replacing of an edge from R by an incoming edge of \mathbf{p} with the same origin,
- 2) replacing an incoming edge of \mathbf{r} which belongs to the path from \mathbf{p} to \mathbf{r} ,
- 3) replacing an incoming edge of \mathbf{r} which belongs to the cycle.

Suppose that at most two such consecutive flips do not increase the number of edges in cycles. Then some new spanning subgraph has maximal non-trivial tree.

Proof. In view of Lemma 7, suppose that R has non-trivial trees.

Further consideration is necessary only if at least two vertices of level L belong to distinct trees of R with distinct roots.

Let the edge $\bar{b} = \mathbf{b} \rightarrow \mathbf{r} \in T$ belongs to the path from R of the maximal length L with beginning in \mathbf{p} . Suppose $\bar{c} = \mathbf{c} \rightarrow \mathbf{r} \in H$. There exists also an edge $\bar{a} = \mathbf{a} \rightarrow \mathbf{p}$ that does not belong to R because Γ is strongly connected and \mathbf{p} has no incoming edge in R . Let $\bar{w} = \mathbf{a} \rightarrow \mathbf{d}$ belongs to R .



Let us consider the path from \mathbf{p} to \mathbf{r} of maximal length L in T . Our aim is to increase the maximal level with an extension of the tree T much more than the maximal level of vertex of other trees from R . We plan to use the three aforesaid ways. If one of the ways does not succeed, let us go to the next, assuming that the situation in which the previous fails, and excluding the successfully studied cases. We can use sometimes two changes together. Let us begin with

1) Suppose first $\mathbf{a} \notin H$. If \mathbf{a} belongs to the path in T from \mathbf{p} to \mathbf{r} then a new cycle with part of the path and the edge $\mathbf{a} \rightarrow \mathbf{p}$ is added to R extending the number of vertices in its cycles in spite of the condition of lemma. In the opposite case the level of \mathbf{a} is $L+1$ in the new spanning subgraph and the vertex \mathbf{r} is the root of the new tree containing all vertices of maximal level (the vertex \mathbf{a} or its ancestors in R).

So let us assume $\mathbf{a} \in H$ and suppose $\bar{w} = \mathbf{a} \rightarrow \mathbf{d} \in H$. In this case the vertices \mathbf{p} , \mathbf{r} and \mathbf{a} belong to a cycle H_1 obtained by removing \bar{w} and adding \bar{a} . We denote by R_1 new spanning subgraph obtained. So we have the cycle $H_1 \in R_1$ instead of $H \in R$. If the length of the path from \mathbf{r} to \mathbf{a} in H is r_1 then H_1 has length $L + r_1 + 1$. A path from \mathbf{r} to \mathbf{d} of the cycle H remains in

R_1 . Suppose that its length is r_2 . So the length of the cycle H is $r_1 + r_2 + 1$. The length of the cycle H_1 is not greater than the length of H because by the condition of lemma the number of edges in the cycles of the spanning subgraph could not grow. So $r_1 + r_2 + 1 \geq L + r_1 + 1$, whence $r_2 \geq L$. If $r_2 > L$, then the length r_2 of the path from \mathbf{d} to \mathbf{r} in a tree of R_1 (as well as the level of \mathbf{d}) is greater than L . The tree containing \mathbf{d} (or some other ancestor of \mathbf{r} in a tree of R_1) is the desired unique maximal tree.

So we can assume for further consideration that $L = r_2$ and $\mathbf{a} \in H$. Analogously, for any vertex of maximal level which belongs to a tree whose root is in H and has an incoming edge starting at \mathbf{a}_1 , the proof can be reduced to the case $\mathbf{a}_1 \in H$ and $L = r'_2$ for the corresponding value r'_2 .

2) Suppose that the set of outgoing edges of the vertex \mathbf{b} is not a bunch. So one can replace in R the edge \bar{b} from the vertex \mathbf{b} by an edge \bar{v} from \mathbf{b} to a vertex $\mathbf{v} \neq \mathbf{r}$.

The vertex \mathbf{v} could not belong to T because in this case a new cycle is added to R , but the number of vertices in the cycles of the spanning subgraph could not grow.

If the vertex \mathbf{v} belongs to another tree of R but not to the cycle H , then T is a part of a new tree T_1 with a new root of a new spanning subgraph R_1 and the path from \mathbf{p} to the new root has a length greater than L . Therefore the tree T_1 is the unique maximal tree in R_1 .

If \mathbf{v} belongs to some cycle $H_2 \neq H$ in R , then together with replacing \bar{b} by \bar{v} , we also replace the edge \bar{w} by \bar{a} . So we extend the path from \mathbf{p} to the new root \mathbf{v} of H_2 at least by the edge $\bar{a} = \mathbf{a} \rightarrow \mathbf{p}$ and there is a unique maximal tree of level $L_1 > L$ which contains either the vertex \mathbf{d} or some of its ancestors in the old spanning subgraph R .

Now it remains only the case when \mathbf{v} belongs to the cycle H . The vertex \mathbf{p} also has level L in new tree T_1 with root \mathbf{v} . The only difference between T and T_1 (just as between R and R_1) is the root and the incoming edge of this root. The new spanning subgraph R_1 has the same number of vertices in their cycles just as R . Let r'_2 be the length of the path from \mathbf{d} to $\mathbf{v} \in H$.

For the spanning subgraph R_1 , one can obtain $L = r'_2$ just as it was done in the step 1) for R . From $\mathbf{v} \neq \mathbf{r}$ follows $r'_2 \neq r_2$, though $L = r'_2$ and $L = r_2$.

So for further consideration suppose that the set of outgoing edges of the vertex \mathbf{b} is a bunch to \mathbf{r} .

3) The set of outgoing edges of the vertex \mathbf{c} is not a bunch because \mathbf{r} has another bunch from \mathbf{b} in virtue of the lemma condition.

Let us replace in R the edge \bar{c} by an edge $\bar{u} = \mathbf{c} \rightarrow \mathbf{u}$ such that $\mathbf{u} \neq \mathbf{r}$. The vertex \mathbf{u} could not belong to the tree T because one has in this case a cycle with all vertices from H and some vertices of T whence its length is greater than $|H|$ and so the number of vertices in the cycles of a new spanning subgraph grows in spite of the assumption of lemma.

If the vertex \mathbf{u} does not belong to T , then the tree T is a part of a new tree with a new root, The path from \mathbf{p} to the new root is extended at least by a part

of H starting at the former root \mathbf{r} . The new level of \mathbf{p} therefore is maximal and greater than the level of any vertex in another tree.

Thus anyway we obtain a spanning subgraph with one non-trivial maximal tree.

Lemma 9 *Let any vertex of an AGW graph Γ has not two incoming bunches. Let R be a spanning subgraph of Γ and let its tree T with the root \mathbf{r} on a cycle H have all vertices of maximal level L and one of them is the vertex \mathbf{p} . We consider the three flips of Lemma 8. Suppose that at most two such consecutive flips do not increase the number of edges in cycles.*

Then coloring R with some color α makes the pair $\mathbf{p}\alpha^{L-1}, \mathbf{p}\alpha^{L+|H|-1}$ stable.

Proof. By Lemma 8, Γ has a spanning subgraph R such that all vertices of maximal positive level L belong to one tree of R . Let us give to the edges of R the color α and color the remaining edges of Γ by other colors arbitrarily.

By Lemma 1, in a strongly connected transition graph for every word s and F -clique F of size $|F| > 1$, the set Fs also is an F -clique of the same size and for any state \mathbf{p} there exists an F -clique F such that $\mathbf{p} \in F$.

In particular, some F -clique F has a non-empty intersection with the set N of vertices of maximal level L . The set N belongs to one tree, whence by Lemma 6 this intersection has only one vertex. The word α^{L-1} maps F on an F -clique F_1 of size $|F|$. One has $|F_1 \setminus C| = 1$ because any sequence of length $L-1$ of edges of color α in any tree of R leads to the cycle C . For the set N of vertices of maximal level holds $N\alpha^{L-1} \not\subseteq C$. So $|N\alpha^{L-1} \cap F_1| = |F_1 \setminus C| = 1$, $\mathbf{p}\alpha^{L-1} \in F_1 \setminus C$ and $|C \cap F_1| = |F_1| - 1$.

Let the integer m be a common multiple of the lengths of all considered cycles colored by α . So for any \mathbf{r} in C as well as in $F_1 \cap C$ holds $\mathbf{r}\alpha^m = \mathbf{r}$. Let F_2 be $F_1\alpha^m$. We have $F_2 \subseteq C$ and $C \cap F_1 = F_1 \cap F_2$.

Thus the two F -cliques F_1 and F_2 of size $|F_1| > 1$ have $|F_1| - 1$ common vertices. So $|F_1 \setminus (F_1 \cap F_2)| = 1$, whence by Lemma 2, the pair of states $\mathbf{p}\alpha^{L-1}$ from $F_1 \setminus (F_1 \cap F_2)$ and \mathbf{q} from $F_2 \setminus (F_1 \cap F_2)$ is stable. Evidently that $\mathbf{q} = \mathbf{p}\alpha^{L+m-1}$.

Theorem 2 [Trahtman 09] *Every AGW graph has a synchronizing coloring.*

Theorem 3 [Trahtman 08a] *Let every vertex of a strongly connected directed graph Γ have the same number of outgoing edges. Then Γ has synchronizing coloring if and only if the greatest common divisor of lengths of all its cycles is one.*

The goal of the following lemma is to reduce the complexity of the algorithm.

Lemma 10 *Let Γ be an AGW graph having two cycles C_u and C_v either with one common vertex \mathbf{p}_1 or with a common sequence $\mathbf{p}_1, \dots, \mathbf{p}_k$, such that all incoming edges of \mathbf{p}_i form a bunch from \mathbf{p}_{i+1} ($i < k$). Let $u \in C_u$ and $v \in C_v$ be the edges of the cycles leaving \mathbf{p}_1 . Let T be a maximal subtree of Γ whose root is \mathbf{p}_1 and whose edges are the union of C_u and C_v except u and v .*

Then the subtree T obtained by adding one of the edges u or v turns in spanning subgraph with one maximal tree.

Proof. Let us add to T either u or v and then find the maximal levels of vertices in both cases. The vertex \mathbf{p}_i for $i > 1$ cannot be the root of a tree. If some tree of spanning subgraph with some vertex of maximal level has the root \mathbf{p}_1 then in both opportunities the lemma holds. If some tree of the spanning subgraph with some vertex of maximal level has its root only on C_u then let us choose the addition of v . In this case the level of the considered vertex is growing. The new tree with root \mathbf{p}_1 is the unique maximal tree. In the case of root in C_v let us add u .

3 The algorithm for synchronizing coloring

Let us start with an arbitrary coloring of an *AGW* graph Γ with n vertices and constant outdegree d . The considered d colors define d spanning subgraphs of the graph.

We keep images of vertices and colored edges of a generic graph by any transformation and homomorphism.

If there exists a loop in Γ around a state \mathbf{r} then let us color the edges of a tree whose root is \mathbf{r} with the same color as the color of loop. The other edges may be colored arbitrarily. The coloring is synchronizing [Adler et al. 77].

In the case of two incoming bunches of a some vertex the origins of these bunches form a stable pair by any coloring (Lemma 3). We merge both vertices in the homomorphic image of the graph (Theorem 1) and obtain according the theorem a new *AGW* graph of the size $|\Gamma| - 1$.

The linear search of two incoming bunches and of loop can be made at any stage of the algorithm.

Find the parameters of the spanning subgraph: levels of all vertices, the number of vertices (edges) in cycles, for any vertex let us keep its tree and the cycle attached to the root of the tree. We form the set of vertices of maximal level and choose from the set of trees a tree T containing a vertex \mathbf{p} of maximal level. This step is linear and used by any recoloring step.

1) If there are two cycles with one common vertex then we use the Lemma 10 and find a spanning subgraph R_1 such that any vertex \mathbf{p} of maximal level L belongs to one tree with root in a cycle H . Then after coloring edges of R_1 by color α we find a stable pair $\mathbf{q} = \mathbf{p}\alpha^{L-1+|H|}$ and $\mathbf{s} = \mathbf{p}\alpha^{L-1}$ (Lemma 9) and go to the step 3). The search of a stable pair is linear and therefore the algorithm in this case is quadratic.

2) Let us consider now the three replacements from Lemma 8 and find the number of edges of the cycles and other parameters of the spanning subgraph of the given color. If the number of edges in the cycles is growing, then the new spanning subgraph must be considered and the new parameters of the subgraph must be found. In the opposite case, after at most $3d$ steps, by Lemma 8, there exists a unique tree.

Suppose the edges of R_1 are colored by color α . Then the vertices $\mathbf{q} = \mathbf{p}\alpha^{L-1+|H_1|}$ and $\mathbf{s} = \mathbf{p}\alpha^{L-1}$ by Lemma 9 form a stable pair.

3) Let us finish the coloring and find the subsequent stable pairs of the pair (s, q) using appropriate coloring. Then we go to the homomorphic image Γ_i/ρ (Theorem 1) of considered graph Γ_i (with a $O(|\Gamma_i|m_id)$ complexity where m_i is the size of the graph Γ_i). Then we repeat the procedure with the new graph Γ_{i+1} which has a smaller size. So the overall complexity of this step of the algorithm is $O(n^2d)$ in majority of cases and $O(n^3d)$ if the number of edges in cycles grows slowly, m_i decreases also slowly, loops do not appear and the case of two ingoing bunches emerges rarely (the most worst case).

Let $\Gamma_{i+1} = \Gamma_i/\rho_{i+1}$ on some stage $i + 1$ have synchronizing coloring. For every stable pair q, p of vertices from Γ_i there exists a pair of corresponding outgoing edges that reach either another stable pair or one vertex. This pair of edges is mapped on one image edge of Γ_{i+1} . So let us give the color of the image to preimages and obtain on this way a synchronizing coloring of Γ_i . This step is linear. So the overall complexity of the algorithm is $O(n^3d)$ in the most worst case and $O(n^2d)$ in most cases. The space complexity is quadratic.

3.1 Check the necessary conditions of synchronizing coloring

The algorithm is based on the Theorem 3. One must check the existence of sink strongly connected component S and check the condition on gcd in S .

Let us use the linear algorithm of finding of strongly connected component SCC [Aho et al. 74], [Tarjan 72]. Then we mark all SCC having outgoing edges to others SCC . If only one SCC does not be marked then sink exists and belongs to this SCC H . In opposite case the synchronizing coloring does not exist.

Then let us study SCC H . Let p be a vertex from H . Suppose $d(p) = 1$. For an edge $r \rightarrow q$ where $d(r)$ is already defined and $d(q)$ is not defined suppose $d(q) = d(r) + 1$. If $d(q)$ is defined let us add the difference $abs(d(q) - 1 - d(r))$ to the set D and count the gcd of the integers from D . If $gcd = 1$ the graph has synchronizing coloring. If after checking all edges from H the $gcd \neq 1$ the answer is negative.

The verifying of the necessary conditions of synchronizing coloring is linear.

References

- Adler et al. 77. R.L. Adler, L.W. Goodwyn, B. Weiss. Equivalence of topological Markov shifts, *Israel J. of Math.*, 27, 49-63, 1977.
- Adler, Weiss 70. R.L. Adler, B. Weiss. Similarity of automorphisms of the torus, *Mem-oirs of the Amer. Math. Soc.*, 98, Providence, RI, 1970.
- Aho et al. 74. A. Aho, J. Hopcroft, J. Ullman. The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
- Béal, Perrin 08. M.P. Béal, D. Perrin. A quadratic algorithm for road coloring. *arXiv:0803.0726v2 [cs.DM]*.
- Bauer et al. 07. T. Bauer, N. Cohen, A.N. Trahtman. The visualization of digraph based on its structure properties. RTAGS, Ramat-Gan, Israel, 2007.
- Budzban et al. 0. G. Budzban, A. Mukherjee. A semigroup approach to the Road Coloring Problem. *Probability on Alg. Structures. Contemporary Mathematics*, 261, 195-207, 2000.

- Carbone 01. A. Carbone. Cycles of relatively prime length and the road coloring problem. *Israel J. of Math.*, 123, 303-316, 2001.
- Černý 64. J. Černý. Poznámka k homogeným experimentom s konečnými automatami, *Math.-Fyz. Čas.*, 14(1964), 208-215.
- Culik et al. 01. K. Culik II, J. Karhumäki, J. Kari. A note on synchronized automata and Road Coloring Problem. Developments in Language Theory (5th Int. Conf., Vienna, 2001), *Lecture Notes in Computer Science*, 2295, 175-185, 2002.
- Friedman 1990. J. Friedman. On the road coloring problem. *Proc. of the Amer. Math. Soc.*, 110, 1133-1135, 1990.
- Gocka et al. 98. E. Gocka, W. Kirchherr, E. Schmeichel, A note on the road-coloring conjecture. *Ars Combin.* 49, 265-270, 1998.
- Hegde et al. 05. R. Hegde, K. Jain, Min-Max theorem about the Road Coloring Conjecture. *EuroComb 2005, DMTCS proc.*, AE, 279 - 284, 2005.
- Jonoska et al. 98. N. Jonoska, S. A. Karl, A molecular computation of the road coloring problem. *DNA Based Computers II, DIMACS Series in DMTCS*, 44, 87-96, 1998.
- Jonoska et al. 1995. N. Jonoska, S. Suen. Monocyclic decomposition of graphs and the road coloring problem. *Congressum numerantium*, 110, 201-209, 1995.
- Kari 01. J. Kari. Synchronizing finite automata on Eulerian digraphs. *Lect. Notes in Comp. Sci.*, Springer, 2136, 432-438, 2001.
- Mateescu, Salomaa 99. A. Mateescu, A. Salomaa, Many-Valued Truth Functions, Černý's Conjecture and Road Coloring. *Bull. of European Ass. for TCS*, 68, 134-148, 1999.
- O'Brien 81. G.L. O'Brien. The road coloring problem. *Israel J. of Math.*, 39, 145-154, 1981.
- Rauff 09. J. V. Rauff, Way back from anywhere: exploring the road coloring conjecture. *Math. and Comput. Education*. 01, 2009.
- Perrin et al. 92. D. Perrin, M.P. Schützenberger. Synchronizing prefix codes and automata, and the road coloring problem. In *Symbolic Dynamics and Appl., Contemp. Math.*, 135, 295-318, 1992.
- Tarjan 72. R.E. Tarjan, Depth first search and linear graph algorithms. *SIAM J. Comput.*, 1(1972), 146-160.
- Trahtman 06. A.N. Trahtman, A package TESTAS for checking some kinds of testability. Proc. of 7-th CIAA. Tours, 2002, 223-227, *Lect. Notes in Comp. Sci.*, 2608(2003), 228-232.
- Trahtman 07. A.N. Trahtman, A subquadratic algorithm for road coloring. *arXiv:0801.2838 v1 [cs.DM]*.
- Trahtman 08a. A.N. Trahtman. Synchronizing Road Coloring. *5-th IFIP WCC-TCS*, Springer, 273(2008), 43-53.
- Trahtman 08b. A.N. Trahtman. The Road Coloring and Cerny Conjecture. *Proc. of Prague Stringology Conference*. 2008, 1-12.
- Trahtman 09. A.N. Trahtman The road coloring problem. *Israel Journal of Math.*, 172(1), 2009, 51-60